Requirements Eng (2017) 22:1–26
DOI 10.1007/s00766-015-0233-3

ORIGINAL ARTICLE

The role of distances in requirements communication: a case study

Elizabeth Bjarnason[1] · Helen Sharp[2]

Received: 8 October 2014 / Accepted: 14 July 2015 / Published online: 26 July 2015

Abstract Requirements communication plays a vital role in development projects in coordinating the customers, the business roles and the software engineers. Communication gaps represent a significant source of project failures and overruns. For example, misunderstood or uncommunicated requirements can lead to software that does not meet the customers' requirements, and subsequent low number of sales or additional cost required to redo the implementation. We propose that requirements engineering (RE) distance measures are useful for locating gaps in requirements communication and for improving on development practice. In this paper, we present a case study of one software development project to evaluate this proposition. Thirteen RE distances were measured including geographical and cognitive distances between project members, and semantic distances between requirements and testing artefacts. The findings confirm that RE distances impact requirements communication and project coordination. Furthermore, the concept of distances was found to enable constructive group reflection on communication gaps and improvements to development practices. The insights reported in this paper can provide practitioners with an increased awareness of distances and their impact. Furthermore, the results provide a stepping stone for further research into RE distances and methods for improving on software development processes and practices.

Keywords Case study · Communication · Requirements · Testing · Measurements · Agile · Distances

1 Introduction

Developing software is a knowledge intense activity where requirements communication plays a vital role in producing a successful product [8, 20, 31, 37]. The customer requirements and expectations need to be communicated to and correctly understood by the development project members [37]. Failure to do so increases the risk of producing a different product from the one the customer expects and can also increase the time and the effort required to achieve the desired product quality [8]. The interaction and communication between individuals and teams plays a vital role in coordinating and aligning the various project activities towards the same goal [8, 43], i.e. to produce a software product that matches the customers' requirements. Testing activities ensure that the released software matches the requirements and the customer expectations. However, this requires coordination and alignment of the RE and the testing activities in which human-to-human communication plays a vital role [10]. This communication can also be facilitated by software artefacts [23]. The structure and quality of these artefacts then influence the alignment between RE and testing [10].

Methods for mapping and improving communication paths by considering the requirements flow have been used to identify issues such as bottlenecks and missing communication between key roles [68]. Requirements communication has also been researched using social network

✉ Elizabeth Bjarnason
elizabeth.bjarnason@cs.lth.se

Helen Sharp
helen.sharp@open.ac.uk

[1] Department of Computer Science, Lund University, Box 118, SE-221 00 Lund, Sweden

[2] Centre for Research in Computing, The Open University, Walton Hall, Milton Keynes, Buckinghamshire MK7 6AA, UK

analysis thereby identifying communication patterns and roles vital for effective requirements-driven collaboration within software development [48, 49].

We propose that distances are important factors that affect the quality and effectiveness of the *requirements communication*, and thus the coordination of requirements throughout a development project from requirements definition to testing. In our previous work we have identified an empirically based theory of distances that states that the effort required to coordinate a project is affected by distances or 'differences in position or level between entities' [12], e.g. people or artefacts. The theory includes a set of *requirements engineering (RE) distances* for the alignment of RE and testing (RET) activities. This set includes people-related distances, e.g. geographical and cognitive, and artefact-related distances, e.g. semantic.

Our theory indicates that a closer integration of RE and testing over one or more distances can support increased communication of requirements within software development and thus improve the alignment between these two activities and thus also the improve project coordination. For example, shorter geographical and cognitive distances between involved roles may facilitate and improve communication, thereby reducing communication gaps and avoiding designing tests based an incorrect understanding of requirements.

For this reason, we are interested in understanding how distances affect requirements communication and if the concept of distance can enable project members to identify and mitigate potential communication gaps, e.g. by an increased awareness of geographical or cognitive distances between the RE and testing roles. *Group reflection* is a method applied in project retrospectives (a.k.a. post-mortem reviews) that can support software process improvement through organisational learning [74]. Group reflection may be strengthened by providing a group with objective data [13], thereby avoiding reflecting on and making improvement decisions based on subjective and incorrect information [41].

In this paper, we report on a case study of an agile development project for which RE distances were measured and presented to and discussed with the project members. The main objectives of the study was to gain a deeper understanding of how distances affect requirements communication and to explore how the concept of distances can be used to support project teams in improving on their development practices. For this reason the following three research questions were defined concerning the RE distances for RET, identified in our previous theoretical work:

RQ1 How do RE distances affect the communication of requirements between RE and Testing activities?

RQ2 How can the concept of RE distances support project teams in reflecting on their requirements communication?

RQ3 How can RE distances and the use of them be enhanced to improve the communication between requirements and testing activities?

These questions were investigated for an ongoing development project through interviews, questionnaires, and ethnographically informed observations [63]. As part of the study, distance measures were designed and applied in order to assess the RE distances within the case project. These measurements were presented visually in a project team focus group as a stimulus to discuss the various distances found.

The rest of this paper is structured as follows; Sect. 2 describes our previous research into RE distances, while Sect. 3 describes related work including RET alignment. The case in which the method was evaluated is presented in Sect. 4, while the applied research method including the design of the distance measures is described in Sect. 5. The measurement instruments are presented in Sect. 6. The research results including the measures for the case project are reported in Sect. 7 and discussed in Sect. 8. Finally, we conclude in Sect. 9 by summarising the findings and outlining future work.

2 Requirements engineering distances

The set of distances investigated in this study is based on the set of requirements engineering (RE) distances identified through previous studies, including one that investigated the coordination and alignment of RE and testing (RET) activities, see Table 1. The concept of distances was initially explored through a systematic mapping study [9] of the use of the term *distance* between RE and later software development activities. The resulting map contained an initial set of RE distances. The challenges and practices of aligning RE and testing activities were investigated through an interview study [12], see Sect. 3.2. This empirical data were analysed against the framework of RE distances obtained from the mapping study and resulted in a refined set of RE distances for RET alignment. In this paper, we investigate and validate this set of RE distances.

2.1 RE distances

The full systematic map of RE distances contains 53 peer-reviewed papers in which 13 different RE distances were found [9]. Eight of the found distances were between people: *geographical*, *temporal*, *socio-cultural*, *cognitive*, *psychological*, *opinion*, *power* and *organisational*. Four

Table 1 The RE distances for RET, derived from [12] and evaluated in this study

Category	Type of distance and example	Between
People	*D1 Geographical distance* Physical distance of desks For example, a physical distance between a product owner and the testers often has a negative effect on the frequency and ease of communication of requirements, e.g. clarifications, detecting misunderstandings and conflicts	Roles related to requirements and testing
	D2 Organisational distance Distance between organisational units For example, when stakeholders and project members are from different parts of an organisation there may be differences in objectives, which cause conflicts and misunderstanding of the requirements	
	D3 Psychological distance Perceived effort to communicate For example, a tester may be reluctant to ask for requirements clarification from people for which it requires a lot of effort to communicate with	
	D4 Cognitive distance Difference in knowledge For example, differences in domain knowledge between a product owner and the development team can lead to differences in understanding of a requirements change	
Artefacts	*D5 Adherence distance* Difference between documented content and perception of agreement or reality For example, there may a difference between the produced software and the specified requirements	Artefact and reality
	D6 Semantic distance Difference in meaning For example, when there is a difference in meaning between the documented requirements and the test cases there is not full test coverage of the requirements	Artefacts
	D7 Navigational distance Effort to navigate For example, for test cases that are not linked to requirements through traces or similar document structures (long navigational distance) activities like impact analysis, test coverage etc. require more effort to perform.	
Activities	*D8 Temporal distance* Time between performing activities For example, a short time (a few weeks) between defining, verifying and validating a user story and its acceptance criteria (requirements) enables a product owner to catch and adjust requirements misunderstandings early on	Activities

distances were between artefacts, namely *semantic*, *similarity*, *syntactic* and *impact*. While finally, one distance concerns the *adherence* between an artefact and reality.

In general, long distance between people has been found to have a negative effect on communication and collaboration within projects, e.g. delays and misunderstandings. However, there are unexplained contradictory findings that indicate that in certain contexts the delay in communication over geographical distance is decreased [72], indicating that there are unknown factors at play. The contradictions may be explained by the effect that different practices have on the characteristics of the communication channel that is used, i.e. formal or informal, synchronous or asynchronous. Berntsson Svensson et al. [7] found that the preference of communication channel and mechanism varies between distributed and co-located coordination. Face-to-face communication was found to be particularly important for the local context and to provide a faster speed of communication in the global context. In contrast, technical-based communication in particular through software architecture was identified as the preferred channel for the investigated software-product line development case, i.e. for large and rich amounts of information.

Although there is less research into distances between artefacts it is an interesting area for future RE research since it is a tangible form of communication between people. It has been suggested as applicable in the context of requirements change and traceability. For example, the distance between the previous and a changed version of an artefact can be used to assess the impact of the change [14, 40]. Distance between RE artefacts and artefacts of later development activities, e.g. design and testing, could potentially be used to measure coverage and consistency between requirements specifications and other artefacts such as design and test specifications, and source code. This potential is investigated through the case study reported in this paper.

The majority of the papers included in the map (42 of 53) were within global software development where geographical, temporal and socio-cultural distances [1] have been found to have a large impact on the development process. Even for co-located development a physical distance of 25 m, i.e. within the same office building, has been found to drastically reduce the frequency of communication between engineers [2]. Our previous studies confirm this and thereby highlight the importance of further research into distances within the co-located context.

2.2 RE distances for RET

In this study we evaluate a set of previously identified RE distances for the alignment of RE and testing (RET) [12]. In that work a structured analysis was performed of interviews from one company (company A, see 0) against the set of (general) RE distances derived from our systematic mapping study (see previous section). This yielded a set of eight distances relevant to the coordination and alignment of requirements and testing activities. These RE distances for RET are: (D1) *geographical*, (D2) *organisational*, (D3) *psychological* and (D4) *cognitive* distances between people; (D5) *adherence* distances to artefacts, (D6) *semantic* and (D7) *navigational* distances between artefacts, and (D8) *Temporal* distance between activities, detailed in Table 1.

3 Related work

3.1 Requirements communication

Communication and coordination of requirements is a challenge within software development in general [31] and in particular within market-driven [42], large-scale [20] and distributed [15, 21, 67] development. The coordination of marketing and development roles within market-driven development is impeded by weak common views on the role of and need for requirements details, lack of common vocabulary, unclear responsibility for requirements specification and analysis, and dependencies on individuals [42]. For large and complex development, these challenges increase and Kraut and Streeter argue that a combination of formal and informal communication is required to cope with uncertainties and changes [43]. For distributed development where informal communication is limited, most of the reported problems are related to communication, e.g. missing context, weak awareness and missing document information [67]. Awareness of one another's work is important since it leads to information sharing and knowledge gain [24]. Lack of knowledge of on-going activities can hinder a correct assessment of the impact of changes, cause misunderstandings about requirements and reduced trust and productivity in a development team [22].

Bridging the *communication gaps* between RE and other development roles and activities, in particular for distributed development, has been identified as an important area for future RE research by Cheng and Atlee [18]. The need for increased insight in this area is further highlighted by Marczak et al. [49] who found that development teams may adhere to very different communication structures than those prescribed in a formal process. Some approaches to bridge geographical distance have been researched including computer-aided support for requirements elicitation and negotiation [16, 21]. Increased awareness of communication paths has also been suggested to improve the information transfer. Kwan et al. [45] proposed visualising dependencies between requirements and the people working on them. Stapel et al. [67] suggested having 'ambassadors' physically present at the different sites and to document fluid information. Marczak et al. [48] found that the information flow is largely controlled by a few *information brokers* who through extensive experience of an organisation and its members can bridge communication gaps.

Matching communication patterns with the technical dependencies between requirements and work items is an approach investigated by several researchers. Cataldo et al. found that the resolution time for a modification request was reduced by a third (on average) when the developers' communication patterns were synchronised with the technical dependencies between their work items [15]. Stapel et al. [68] propose a related approach to managing information flows named FLOW Mapping. FLOW Mapping entails capturing the information needs of a project, developing and implementing a communication strategy covering both formal and informal channels, and then monitoring and measuring adherence to this strategy.

In this study we explore an alternative approach in bridging communication gaps. Rather than map and analyse the information flow or consider known challenges for weak communication and coordination we investigate the concept of distance as an underlying cause of communication gaps. In our previous work on generating an empirically based theory of distances [12] we found indications that when there is a long distance between people and between artefacts this has a negative effect on the communication and coordination. Therefor we now investigate whether distances within a development project can be used to indicate a high risk of communication gaps occurring within the project team and between individuals between which there is a long distance.

3.2 Aligning RE and testing activities

There is a limited amount of research on aligning and coordinating the activities of requirements engineering and testing, i.e. RET alignment, rather, most research tends to focus on one area [4]. Barmi et al. found that most studies of RET are on model-based testing. Only three empirical studies specifically focusing on RET alignment were found (in 2011), and Barmi et al. [4] identified this as one of the main areas for future RET research. Furthermore, Barmi et al. draw the conclusion that although the areas of model-based engineering and traceability are well understood, there is a need for practical approaches and methods for implementing these.

Empirical studies of RET alignment consist of one case study into jointly improving the RE and testing processes by Kukkanen et al. [44], one interview study investigating industrial practices in linking RE and testing by Uusitalo et al. [70] and our own study on RET-related challenges and practices in industry [10, 66]. Kukkanen et al. [44] found that RET alignment can be improved by integrating the requirements and testing processes. The most important aspect in achieving alignment was found to be ensuring that ‘the right information is communicated to the right persons’. The practices implemented to support this were: metrics, tool-supported traceability, a change management process and requirements reviews. Similar and additional RET alignment practices are reported by Uusitalo et al. [70] including a number of practices that increase the communication and interaction between requirements and testing roles, namely early tester participation, traceability policies, considering feature requests from testers, and linking test and requirements people. Our RET study identified 10 categories of challenges and 10 categories of practices, covering a wide range of aspects such as communication, requirements quality, organisation, processes and tools [10].

Traceability has been researched since the beginning of software engineering in 1960s [60]. However, despite its (acknowledged) importance in high-quality development [59, 71] the implementation of this practice remains elusive and challenging [33, 39, 58, 71]. Traceability between requirements and other development artefacts can increase the product quality [57, 71] by supporting impact analysis [23, 33, 44, 57, 70, 71] lowering of testing and maintenance cost [44, 71], and increasing test coverage [70, 71]. However, challenges with traceability have also been reported, e.g. by Cleland-Huang. These challenges include artefact volatility, informal processes, lack of clear responsibilities, communication gaps, insufficient time and resources, low insight into cost-benefit of tracing, and a lack of training [19]. It has been suggested that the cost of traceability can be reduced by automatic or semi-automatic recovery of traces [26, 36, 46] or by tracing at a higher abstraction level, e.g. user scenarios, thereby reducing the number of traces [56]. In the context of our work, it is interesting to note that the traceability specialists Gotel and Finkelstein express that a particular concern in improving requirements traceability is the need to facilitate informal communication with those responsible for specifying and detailing requirements [33].

Model-based testing (MBT) is a large research field within which many formal models and languages for representing requirements have been suggested [27]. MBT has issues with practical applicability [53, 55, 73], but there are exceptions. Hasling et al. [35] and Nebut et al. [55] report on experiences from applying MBT by generating system test cases from requirements expressed in UML. The main benefits of MBT are increased test coverage [35, 55], improved requirements quality [35] and increased testing productivity [34]. However, the formal representation of requirements requires special competence to produce [55] and understand, and poses challenges in communicating with, e.g. business roles [47]. In addition, the risk of errors in the models needs to be considered [35]. An alternative to formal models is the use of scenario-based models for high-level requirements and test cases for detailed requirements information. This approach has been proposed by Regnell and Runeson [61], Regnell et al. [62] and Melnik et al. [51] and is often applied in agile development [59]. Melnik et al. [51] found that this approach in combination with executable acceptance test cases is straightforward to implement and breeds a testing mentality. Similar positive experiences are reported by Martin et al. [50].

To summarise, the research on RET alignment ranges from human interaction between RE and testing roles to techniques and tools for connecting and coordinating the requirements and testing artefacts. Both of these concern requirements communication either direct communication between people or indirect communication via documentation. The set of RE distances evaluated in this study cover both of these aspects although with an emphasis on the direct communication between people.

3.3 Using measurements for group reflection

Brede Moe et al. [54] propose the use of a *team radar* for supporting agile development teams in discussing and identifying how to improve on their teamwork. The team radar visualises 5 qualitatively measured factors derived from an empirically based theory on teamwork challenges, namely shared leadership, team orientation, redundancy, learning and autonomy. The approach was found to provide a common vocabulary for the practitioners and the researchers to discuss teamwork and was useful to practitioners in identifying improvements. Angermo Ringstad et al. [3] extended the *team radar approach* by strengthening the diagnosis step and by introducing action planning in the group reflection step. With these improvements, issues previously not discussed within the team were identified due to the approach of highlighting underlying factors and causes, rather than merely pointing out experienced problems.

There are several similarities between the team radar approach and this study. Both approaches are based on measuring, visualising and presenting *psychometrics* [30] of a development project to its team members with the aim of improving the team's ability to collaborate through increased awareness and team learning. Furthermore, the set of measurements are in both cases based on a theory of factors affecting the issue targeted by the approach, in our case requirements communication and coordination.

4 Case description

A development project within The Open University's IT unit provided the case for this study. The Open University is UK's largest academic institution with more than 240,000 students from all over the world. The IT unit is responsible for the day-to-day management of the university's information systems and in-house development of some systems. The studied project is part of a programme developing a system for student administration and curriculum management to meet the new requirements posed by evolving curriculum needs, changed fees and funding regulations, and subsequent changes to internal business processes. An overview of the case is provided in 0.

The Scrum development method is applied at team and intra-team levels. Each development team consists of a product owner, a requirements analyst, a tester, a number of developers and a scrum master. In addition, there is a project manager responsible for the project to which the team delivers. The product owner represents the business and is responsible for the scope including signing off on acceptance of project deliveries. The requirements analyst is responsible for eliciting and defining the requirements in close collaboration with the product owner and the development team. The scrum master, project manager, developers and testers all take an active part in discussing, and thereby defining, the requirements. Finally, the tester within the team is responsible for verifying that the produced software corresponds to the requirements. The team members of the studied development team are characterised in Table 3.

The project scope is described in definition documents and in agile epics by senior requirements analysts and allocated to one of the four planned system releases. For each release, the requirements analyst for the intended development team details the epics into user stories and acceptance criteria and places these in the team's backlog. Development is performed in 2-week sprints (iterations), and prior to each sprint the user stories in the backlog are prioritised by the product owner and requirements analyst. The user stories with the highest priority are then presented to the development team who estimate them. A set of stories are agreed on for that sprint according to priority and team capacity.

Development of a user story is initiated by a discussion between developers, requirements analyst and tester around requirements and technical details. The requirements analyst discusses uncertainties or questions regarding the user requirements with the product owner as they surface. The tester develops test scripts to verify the agreed requirements. Completed user stories, i.e. developed and successfully verified, are demonstrated to the product owner at the end of the sprint. A retrospective meeting is then also held where the development team reflects on the past sprint and on ways to improve team work practices.

The epics, user stories and acceptance test cases are stored in a central requirements repository with traceability links. The test scripts are stored in another repository with traces to the relevant user story. These test scripts can be viewed from the requirements repository.

Once the development team has delivered accepted functionality, user acceptance and system integration testing is performed by representatives from the business unit and by team-external testers. Any found issue is first analysed by the tester in the development team before the issue is either rejected or agreed to be resolved. The development team tester and the system integration testers belong to the same department.

5 Research method

A case study [64, 65] of an ongoing development project was performed in order to further investigate the previously identified RE distances and to explore the rich fauna of factors which may potentially be impacted by these distances (RQ1). In addition, we wished to investigate how distance measurements can be used to reflect and improve on development practices (RQ2). A formative evaluation was performed to seek feedback that can guide future improvements in order to ensure usability and usefulness of the concept of measuring and visualising distances (RQ3).

The study consisted of four main parts, namely (I) preparations, (II) data collection, (III) evaluation and validation, and (IV) data analysis, see Fig. 1. Each part of the study is described below. The study design, data collection and analysis was mainly performed by Bjarnason, and reviewed and validated by Sharp. In addition, Sharp provided support in the contact with the case organisation and participated in one initial interview and in the focus group session where an iRE profile, i.e. the outcome of the

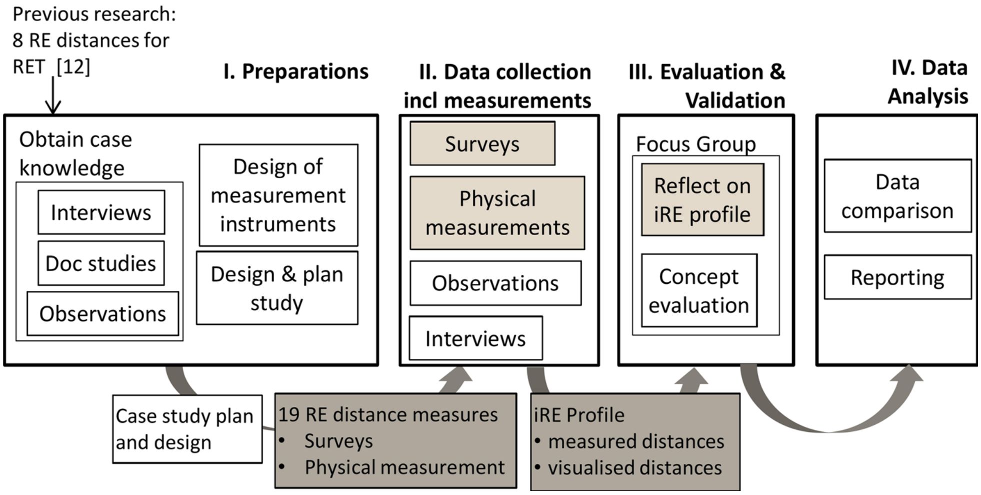

Fig. 1 Overview of the applied research method. The activities that are part of the evaluated concept, i.e. measuring distances, are marked with *grey*; *light grey* for activities, and *dark grey* for artefacts and data

distance measurements, was presented to the development team.

5.1 Preparations

In the initial phase (part I of Fig. 1), the study was designed and planned, and measurement instruments for assessing the RE distances were produced. This required insight into the case, in particular, the roles, artefacts and practices of the studied team.

5.1.1 Obtaining case knowledge

Initial knowledge of the case was obtained through document studies, a semi-structured interview, demonstrations, and observations of the development team prior to performing the measurements (described in Sect. 5.2) One of the authors had an existing relationship with the studied organisation and therefore also some initial documentation and contacts. The researchers studied and discussed these documents and an interview instrument was designed (available on-line, see [11]) to obtain knowledge about the roles, artefacts and activities used for RE and testing. Two managers within the IT development unit agreed to participate in this initial interview. At the managers' suggestion they were interviewed at the same time using an open semi-structured interview format. The managers shared their view of current challenges and good practices and supplied a number of pointers to information and people including access to various development artefacts, e.g. requirements, backlogs, test cases etc.

Insight into development artefacts was obtained through document studies and demonstrations of the artefacts used for testing. The amount and extent of available artefacts and stored information was investigated by studying the requirements and testing artefacts.

Finally, an initial observation of the development team was performed. One researcher was present in the team area for a consecutive period of 3 days at the end of one sprint, including review and planning meetings for the next sprint. The researcher did not interact or disturb the project members, but merely observed how, with whom and about what they interacted. This allowed the researcher to gain familiarity with the team and with their day-to-day work. This insight enabled detailed design of the measurement instruments and of the research method.

5.1.2 Design of the measurement instruments

For each distance, aspects to measure were identified, then for each aspect, measurements and scales were defined. For example, for semantic distance between artefacts, the following aspects were defined: similarity in meaning, abstraction level and coverage. The measurement instruments were implemented as one physical measurement (for geographical distance) and three questionnaires, namely profile, communication and artefact questionnaires (available on-line [11]). The profile and communication questionnaires measure people-related distances, i.e. organisational, cognitive and psychological distance, through self-assessment questions. The artefact questionnaire measures artefact-related distances, i.e. semantic and adherence distance, related to example artefacts. See Sect. 6 for more details on the measurements.

The design was guided by our previous empirical data and related research findings, and by the process and practice of the case project. For example, the measures for cognitive distance include difference in domain knowledge between RE and testing roles since our empirical data indicates that this affects requirements communication. Similarly, the measures for semantic distance include similarity in meaning between the user stories and the acceptance test cases based on information obtained about the case.

The researchers excluded two distances for this cases study, namely navigational (D7) and temporal distance (D8). The case project claimed to apply full traceability, thus yielding a navigational distance of one for all relevant requirements-test case connections. Similarly, since requirements are defined and changed through direct communication within the team it is practically hard to measure temporal distance, i.e. time between defining a requirements and defining a test case for that requirement.

5.1.3 *Design and planning of the study*

The study was designed based on case characteristics with the aim of evaluating the outcome of the measurements. The researchers decided to evaluate the distances by measuring these for the development team during two subsequent sprints, i.e. for a period of 4 weeks. This allowed for studying a full set of development activities including requirements detailing, design, development and testing within a feasible time frame and with a delimited set of requirements.

We decided to evaluate the concept of distance measures through a focus group with the development team. This allowed us to evaluate how well this approach can support a team in reflecting on their requirements communication, while simultaneously validating the measurement instruments and eliciting information regarding the impact of distances.

5.2 Measurements and data collection

The RE distances within the case project were measured by applying the measurement instruments, and additional data relevant to these distances, e.g. issues experienced by the team, were gathered through interviews and observations. All the roles within the team were interviewed and observed. The multiple types of data allowed us to explore the potential impact of distance and to partially validate the measurements by applying triangulation. The measurements were combined and visualised in the iRE profile to present the findings to the development team and support group reflection around the found distances including causes, consequences and relationships between the distances (Table 2).

Table 2 Characteristics of the open university case for which the RE distances were evaluated (this paper) and of company a from which the RE distances were empirically derived (see Sect. 2.2). The cases are here presented together to allow for comparing the two

	Open university	Company A
Type of case	Academic education provider	Software development of embedded products
# People in software development unit	Approx. 150 for IT development (300 for whole IT unit)	125–150
# People in project	Approx. 20	10
Distributed	No	No
Domain/system type	IT: Educational programme management including student services	Computer networking equipment
Source of requirements	In-house	Market driven
Main quality focus	Maintainability	Availability, performance, security
Certification	No	Not for software
Process model	Scrum	Iterative
Duration of project	2–3 years	6–18 months
# Requirements in project	Approx. 800 user stories	100 (10–30 pages of html)
# Test cases in project	Approx. 1300 test cases	Approx. 1000 test cases
Product lines	No	Yes
Open source	No	Yes

5.2.1 *Questionnaires and interviews*

The measurement instruments consisting of the profile, communication and artefact questionnaires were used to assess the RE distances. The people-related distances were measured by administering the communication questionnaire to all team members, while the profile questionnaire was administered to cover all roles within the team. In contrast, the artefact questionnaire used to assess the artefact-related distances was only administered to the product owner, requirements analyst and tester since these are the primary roles involved in detailing the relevant artefacts. For each questionnaire, the targeted respondents were free to choose whether or not to participate. The measurement instrument is described in more detail in Sect. 6.1, while the number and role of the respondents per questionnaire is shown in Table 3.

The profile and artefact questionnaires were administered as semi-structured interviews in order to elicit a richer picture of issues potentially related to distances. For each question, the interviewer ensured that the interviewee understood the question and the scale correctly. Follow-up questions were asked to clarify the interviewee's responses and gather information concerning events contributing to or resulting from each distance. During the interviews the

Table 3 Roles and length of experience for the members of the team included in this case study

Roles	Length of experience in team role (months)	Total length of work experience (years)	Questionnaire			Focus group	
			Profile	Communication	Artefact	Meeting	Individual
Product owner	10	26	1	1	1		1
Requirements analyst	0	28	1	1	1		1
Tester	3	26	1	1	1	1	
4 Developers	8, 9, 9, 0	7, 6, 22, 10	2	4	0	4	
Scrum master	10	26	1	1	0	1	
Project manager	3	25	1	1	0		1

The number of respondents for each questionnaire and number of participants of focus group meeting (joint meeting or individual follow-up) is also given, see Sect. 5.2

answers to the questions were noted by the interviewer on a copy of the questionnaire. The interviews were audio recorded and transcribed, and analysed in the final step of the study.

The communication questionnaire on ease of communication with individual team members was not combined with interviews due to the sensitive nature of the questions. In order to obtain honest answers, the questionnaire was answered privately by each team member and no names or roles were given when presenting the results. Furthermore, there was no need for collecting more information through interviews at this point since a good general insight into team communication had been obtained through observations.

Information on where each team member was seated was obtained through the profile questionnaire and through observations of the team area. The geographical distance between team members was then measured with a tape measure within the team area and through estimates based on a map for desks in the other buildings.

5.2.2 *Constructing the iRE profile*

The iRE profile was constructed by combining the set of measured distances with the aim of supporting the project team in understanding and constructively discussing the outcome of the measurements. An explorative approach was used in designing the presentation and visualisation of the iRE profile. Various calculations of total distance were investigated based on the distance measurements. For example, average, minimum, maximum, sum of pair-wise distances between the data points, and Cartesian difference for multi-dimensional data points. Similarly, various visualisations of distances were explored including radar diagrams, plotting of data points, and graph representations. An example of part of the iRE profile constructed for this case study is shown in Fig. 2. See Sect. 6.2 for a more detailed description of the iRE profile concept.

5.2.3 *Ethnographically informed observations*

An ethnographically informed approach was applied when observing the development team with the purpose of gaining insight into the interactions and day-to-day work practices of the team members. The ethnographical approach entailed seeking to understand the team's work practices apart from the researcher's assumptions about software development [63].

One researcher observed the main team area where the developers, the tester and the scrum master were located during the main data collection period, i.e. for 4 consecutive weeks. The researcher attended team meetings, e.g. the daily stand-up meetings, and observed project members as they worked and in particular when, with how and about what they interacted. The observations were as unobtrusive as possible and questions were only asked to seek clarification of used terminology or actions, and not to participate in team discussions.

The set of RE distances provided a 'protocol' that supported the observer in taking particular note of activities and interactions potentially related to these. For example, for geographical distance, physical location and movements in the team area were noted. For cognitive distance, explicit explanations of domain knowledge provided to newer team members were noted. Extensive field notes were made on interactions in the team area, status and information shared during meetings, and individual activities.

5.3 Evaluation: focus group with development team

The iRE profile was presented to the project team at a focus group session [64] to elicit the practitioners' views on the relevance and validity of the distance measurements regarding their communication and coordination of requirements within the project and, in particular, towards testing. For each distance type, the relevant parts of the iRE

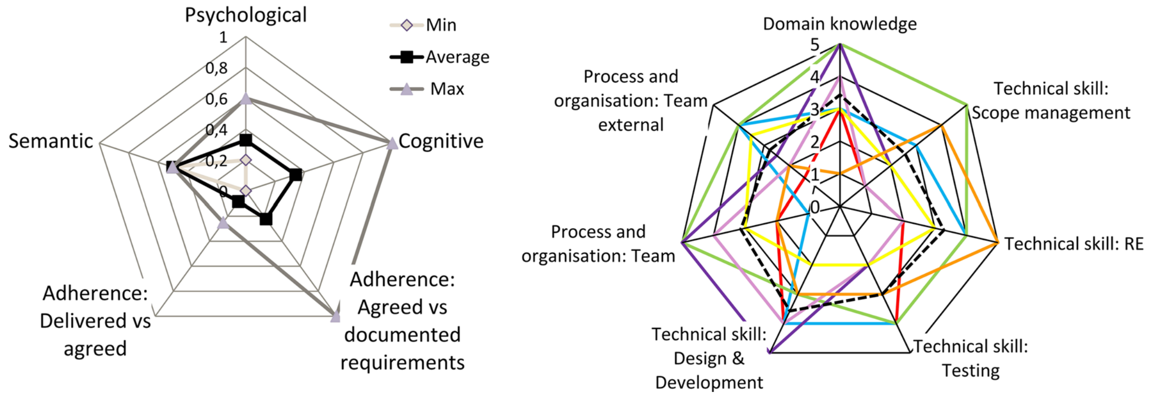

Fig. 2 Examples of radar diagrams used in the focal group session to provide a visual overview of a sub-set of the distance measurements within iRE profile of the case project. The *left-hand diagram* shows the normalised minimum, maximum and average values for several distances within the whole project team. This enable quickly grasping for which distances there might be gaps (large distances) within the team. The *right-hand diagram* shows the level of cognition for each member and the average for all. This allows for considering distances between individuals. At the focus group session, information of which role *each line* represented was given, after obtaining permission from each project member. This information has been removed here for reasons of anonymity

profile were shown and the project members shared their observations of potential issues caused by the distance. This allowed for a validation of the obtained measurements while also eliciting knowledge of the impact of the distances on RET alignment.

The focus group was opened with an introduction to RET alignment and the concept of RE distances including an overview of the measured distances. Then for each distance, the obtained measurements were presented followed by an open question about if and how this distance may have an impact on the team's requirements communication and coordination. For measurements with identical scale, or a scale that can be normalised, the distance values were shown using radar diagrams. Figure 2 contains two examples of such diagrams. The left-hand diagram provides an overview of the range of distances within the team by showing the minimum, maximum and average distances between pairs of team members for four aspects of cognitive distance, namely domain knowledge, quality prioritization, technical knowledge, organisational and process knowledge. The right-hand diagram shows the measured levels of cognitive aspects of each individual team members and the average level (black line). With this diagram the participants could consider distances between pairs of project members, e.g. between the product owner and the tester.

When all the distances had been discussed, the participants were asked to reflect individually on issues related to the presented distances. These reflections were written on post-it notes and then shared and discussed. Finally, the participants were asked if and in which way the distance measures were useful and had supported them in reflecting on their requirements communication practices.

The session was audio recorded (after agreement was obtained from the participants), transcribed and summarised. This summary was then distributed to all the team members who were asked to provide feedback if anything was incorrectly described or if they had additional reflections.

The whole team was invited to the session and six of nine team members attended. The content and questions of the session were later covered with the three absent team members through individual semi-structured interviews following the same structure as the meeting. See Table 3 for details on individual team members.

5.4 Data analysis and reporting

The complete set of data gathered was analysed in the final part of the study and reported in this paper. The researchers' experiences of applying the measurement instruments and using them in the focus group were also considered. The data were analysed in two main iterations, one before and one after the focus group meeting where the initial findings were presented. During the initial analysis, the measurements and the interview transcripts were stored in a spread sheet and categorised (or coded) per distance type. For each type of distance, the relevant data were analysed together and compared against the observations. The aim of this analysis was to identify troublesome distances (gaps) and issues experienced within the development team, and potential connections between these. These findings were presented at the focus group session.

Similarly, the transcripts of the focus group session were categorised per distance and analysed per distance type, thereby providing triangulation and validation of the outcome of the initial analysis. In addition, the participants' viewpoints on the focus group meeting and the concept of distance were categorised and analysed together with the

Table 4 Overview of the measurements (M1–M8) per distance (D1–D8, see Table 1) and the questionnaire used to collect the needed data

Distance	Measurement aspect		Description	Questionnaire
D1 Geographical	M1	Physical	Physical distance between desks	Profile
D2 Organizational	M2	Home unit in line organisation	Length of path in line organisation chart between two people	
D3 Psychological	M3.1	Uni-directional	Perceived effort to communicate with another person	Communication
	M3.2	Bi-directional effort	Perceived effort to communicate between two people	
D4 Cognitive	M4.1	Domain knowledge	Difference between people's knowledge of system domain	Profile
	M4.2	Technical skill	Differences in competence within technical areas affecting RET alignment	
	M4.3	Process and organisation	Differences in knowledge of project and organisation including processes	
	M4.4	Priorities	Differences in prioritisation around product	
D5.1 Adherence: Delivered versus agreed requirements	M5.1.1	Similarity	Difference between product actual and agreed product behaviour	Artefact
	M5.1.2	Coverage	Difference in coverage between actual and agreed product behaviour	
D5.2 Adherence: Agreed versus documented requirements	M5.2.1	Similarity	Difference in meaning between documented versus agreed requirements	
	M5.2.2	Coverage	Degree of coverage between documented versus agreed requirements	
	M5.2.3	Abstraction	Difference in abstraction level between documented versus agreed requirements	
D6 Semantic: Requirements versus test cases	M6.1	Similarity	Difference in meaning between requirements and testing artefacts	
	M6.2	Coverage	Degree of coverage between requirements and testing artefacts	
	M6.3	Abstraction	Difference in abstraction level between requirements and testing artefacts	
D7 Navigational	M7.1	Requirements to test cases	Number of clicks to navigate from a requirement to the test cases which verifies it	
	M7.2	Test case to requirements	Number of clicks to navigate from a test case to the requirement(s) that is verifies	
D8 Temporal	M8	Requirements—test case definition	Length of time between specifying a requirement and defining a test case for verifying it	

For some distances, several aspects were measured

researchers' experience of applying the measurement instrument. The results of this analysis are reported in Sect. 7, and a full comparative analysis per research questions is reported in Sect. 8.

6 Measuring distances

6.1 The measurement instruments

The measurement instruments assess the previously identified RE distances [12] (see Sect. 2.2) between the artefacts and between the people involved in requirements and testing activities. While some distances are straightforward to measure, others are estimated through questionnaires with self-rating questions. For example, geographical distance (D1) is assessed by measuring the physical distance to walk between two desks, while psychological distance (D3) is measured through a question where team members rate this distance to each other. The measurements are listed in Table 4.

A majority of the distances are complex and are measured for several aspects. For these distances there is one measurement per aspect and subsequently several measurements per distance. For example, five aspects are measured for cognitive distance (D4); one aspect of prioritisation of system quality aspects, and three aspects of differences in knowledge, namely domain knowledge, technical skill and knowledge of process and organisation.

Most of the questions have Likert-type scales with five options. For all of these, a numerical value corresponding to the option is used when normalising, analysing and visualising the corresponding distances. The questionnaire respondents were aware of these numerical scales.

Some distances were directly measured by a question, while others were calculated as the difference between pairs of given responses. For example, psychological distance (D3, M3.1) was measured by each respondent rating how hard it was to communicate with another colleague, *Not hard (1), Some effort required* (2), *Medium effort* (3), *Much effort* (4), *Extremely hard* (5).

In contrast, for the knowledge aspects of cognitive distance (M4.1–M4.3) each participants graded their competence using Benner's [6] five levels of experience, i.e. *Novice* (1), *Advanced beginner* (2), *Competent* (3), *Proficient* (4) and *Expert* (5). The cognitive distance was then measured by calculating the difference between two people's levels of competence.

For the artefact questionnaire, the aspects abstraction (M5.2.3, M6.3) and coverage (M5.1.2, M5.2.2, M6.2) are directional, i.e. the abstraction level of artefact A may be higher or lower than artefact B. For these questions the following scale was used: *Much more (−2)*, *Somewhat more (−1)*, *The same (0)*, *Somewhat less (1)*, *Much less (2)*, and *Can't say.*

The aspect of priority for cognitive distance (M4.4) was assessed with a question on the relative priority of the quality characteristics specified in ISO/IEC 9126-1. The respondent was asked to distribute 30 *resources* over the six quality characteristics. The distance between two people was then assessed by calculating the Cartesian distance between their responses (one value per quality characteristic for each person).

6.2 The iRE profile

We define a project's integrated RE profile for testing, *iRE profile*, to be the combined set of measured distances for the project and the visualisation of these. An iRE profile provides a view of the project's current level of RET integration and is produced by collating the measurements for each distance. The iRE profile was used to present the measurements to the project team at the focus group session, see Sect. 5.3.

A combined view of multiple data points can be provided through the iRE profile, which may support reflection of distances within a group of people. For instance, by presenting the range and average value for a distance, project members at short, long or average distance from each other can be identified. For example, team members at above-average psychological distance compared to the rest of the team could indicate team members who are misunderstood.

Similarly, displaying several distances together can support reflecting on relationship between them. For example, short geographical distance between a requirements engineer and the tester may compensate for a large cognitive distance since this cognitive gap can then be bridge by frequent communication (facilitated by short geographical distance). For distances with the same scale, or scales that can be normalised, such relationships between distances can be facilitated by visualising them together in a radar diagram, see examples in Fig. 2.

Upon re-assessing a project, updated versions of the iRE profile can be compared to assess the effect of the implemented practices. For example, if the implementation of cross-role reviews has affected the previous semantic distance between requirements specifications and test cases.

7 Results

The main results of this study are: the distance measures for the case project, as recorded through the set of instruments introduced in Sect. 5.1.2; and how these distances, or lack of them, relate to issues and opportunities within the project. Each measured distance is presented below (in Sect. 7.1) alongside related data captured for each distance through the observations, the interviews and the focus group. The practitioners' reflections on the relevance and usefulness of the concept of distances are also reported below (Sect. 7.2), followed by a discussion of limitations and threats to validity for these results (Sect. 7.3).

7.1 The iRE profile for the case project (RQ1)

An iRE profile constructed from the distances measured for the project was used to investigate how distances impact the coordination between RE and testing. An overview of the derived profile and the minimum, maximum and average (mean) values for each measurement is shown in Table 5. The obtained values for each type of distance are presented below together with qualitative data from the observations and the interviews.

7.1.1 Geographical distance (M1)

The core team members (scrum master, developers and tester) were co-located in one common team area, while the other team members (product owner, requirements analyst and project manager) were located elsewhere. The project manager was seated on the same floor as the core team. The

Table 5 An overview of the iRE profile for the case project

Measured distance				Min	Average	Max
M1	D1 Geographical (metres)			1.8	76.7	322
M2	D2 Organisational (steps in organisational path)			0	2.6	7
M3.1	D3 Psychological	Person to person (uni-directional)		0.20	0.35	1
M3.2		Between two people (bi-directional)		0.20	0.35	0.60
M4	D4 Cognitive	In total:		0.15	0.29	0.46
M4.1		Domain knowledge		0.00	0.32	0.80
M4.2		Technical skill	In total:	0.17	0.32	0.50
M4.2.1			Scope management	0.00	0.36	0.80
M4.2.2			Requirements engineering	0.00	0.27	0.60
M4.2.3			Testing	0.00	0.23	0.40
M4.2.4			Design and development	0.00	0.23	0.60
M4.3		Process and organisational knowledge	In total:	0.06	0.32	0.56
M4.3.1			Local	0.00	0.38	0.80
M4.3.2			Non-local	0.00	0.25	0.52
M4.4		Priorities		0.03	0.08	0.14
M5.1	D5 Adherence	Agreed versus documented requirements	In total:	0.00	0.24	1.00
M5.1.1			Similarity	0.00	0.00	0.00
M5.1.2			Coverage	0.00	0.00	0.00
M5.1.3			Abstraction level	0.50	0.67	1.00
M5.2		Delivered behaviour versus agreed requirements	In total:	0.00	0.08	0.25
M5.2.1			Similarity	0.00	0.17	0.25
M5.2.2			Coverage	0.00	0.00	0.00
M6	D6 Semantic[a]	In total:		0.3	0.4	0.5
M6.1		Similarity		Roughly the same		
M6.2		Coverage		Somewhat more		
M6.3		Abstraction level		Somewhat more		

The numbers show minimum, maximum and average (mean) of the obtained distance measures (see Sect. 6 for details on how the distances were measured). All values except for geographical and organisational distance are normalised within the range of 0–1

[a] One data point only for the measurements of semantic distance

requirements analyst was located on a different floor in the same building. The product owner was approximately 300 m away in a separate building. The total distance between all team members was 2760 m, while the average distance between each pair of team members was 77 m, see Table 5.

The team was acutely aware of the negative impact of geographical distance and frequently commented on the lack of proximity to the product owner and the requirements analyst. During the interviews several people commented on how the geographical distance causes time delays in obtaining information. As expressed by one team member: ‘the conversation slows down’. For example, quick questions concerning requirements may be postponed and then forgotten, or posed to a team member closer at hand. This then results in proceeding with potentially incomplete or incorrect information about the requirements. One interviewee said: ‘Even being 2 desks away can have a negative impact. It makes a big difference! It [co-location] makes it easy to quickly check details you are unsure about.’

Co-location, i.e. short geographical distance was perceived by the team as enabling them to manage requirements changes in a light-weight manner by relying more on frequent face-to-face communication than on extensive documentation of requirements. As the product owner stated: ‘we get what we expect due to the constant communication.’ The requirements analyst also stated that the geographical distance to the team reduced communication and they attempted to mitigate this in part through documentation. Information concerning requirements is also frequently picked up by the tester from on-going discussions in the team area.

Furthermore, the geographical distance sometimes leads to a lack of coordination. This was expressed in interviews with the product owner and the requirements analyst, and observed when meetings were cancelled, delayed or moved to another meeting room with short notice. Information concerning these changes was shared between the co-located team members but did not always reach the team members outside of the team area.

7.1.2 Organisational distance (M2)

All team members except the product owner belonged to the IT unit. The scrum master and the developers were organised into one department, while the requirements analyst, tester and project manager each reported to separate managers within the IT unit. The maximum organisational distance found within the team was between the developers and the product owner who was from the business unit, i.e. outside of the IT unit, at a total distance of 7 steps up and down the organisational tree. In total there was an organisational distance between each pair of team members of 92 steps.

Two team members described that people from other organisational units can disagree due to different priorities and perspective, e.g. on how and which requirements to implement. The product owner mentioned this for the business unit versus the IT unit, while a developer described a similar situation between the development team and other functions within the IT unit. They both stated that these long organisational distances between units make it infeasible to communicate using the organisational hierarchy for decision making and for resolving disagreements concerning requirements. When two distant organisational units escalate issues to their common manager, who is located at a high level in the organisation, this manager is then often too far removed from the context and day-to-day work of the issue at hand to make an informed decision. Several interviewees had found that escalating decisions in this way causes long delays and miscommunication of information.

When there is a long organisational distance between roles, the team members found that communicating informally or through project meetings was a more direct communication channel and therefore more efficient. For example, the product owner had established direct communication channels by attending various project meetings held by the IT department, including meetings concerning project steering, scope and issue management. Similarly the scrum master described that conflicts with other IT development roles were avoided as far as possible by direct communication and by pro-actively seeking alternative solutions. However, both product owner and scrum master mentioned cases where these more direct communication channels failed to achieve an agreement on, e.g. important user requirements or design issues. When this occurs, the issue is either escalated through the organisational channels with subsequent long delays, or left unresolved.

Furthermore, the organisational distance also caused practical issues with coordinating meeting schedules. The product owner who frequently attends various meetings at the IT department expressed that these often conflict with other meetings within the business unit.

7.1.3 Psychological distance (M3)

The psychological distance between team members was on average short; between *Not hard* (1) and *Some effort required* (2) (on average 1.7 out of 5). However, in some cases, the distance was long, indicating that there is psychological distance between certain members of the team. There were two counts of *Extremely hard* to communicate given by one practitioner, and three counts of *Much effort* by two other team members, see Fig. 3. These values are for the uni-directional distance, i.e. one person's perception of communicating with another. However, the values for bi-directional distance, i.e. the average perception of each pair of team members, are distributed closer to the middle of the scale with most scores for *Some effort* and none on the two highest value options. It is interesting that the psychological distance between two people, as perceived by one of them, is not always reciprocated by the other, i.e. if person A finds it hard to communicate with person B it does not necessarily mean that B finds it hard to communicate with A.

The observations revealed that the communication within the team is good and that there is a strong awareness of the importance of sharing information. For example, information sharing practices were emphasised by several team members at a sprint retrospective. In addition, use of these practices was observed when new team members arrived in the team. For example, information was shared with a new developer by frequently pairing with the more

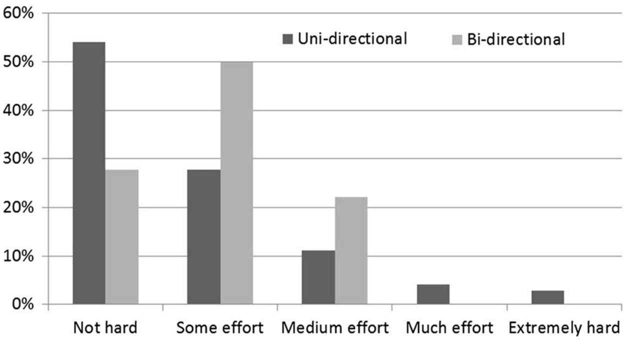

Fig. 3 Percentage of occurrences for uni-directional versus bi-directional psychological distance for each pair of team members

experienced developers. Furthermore, information concerning context and motivation for team practices was spontaneously shared with new team members.

Occasional occurrences of communication difficulties were observed, even though in general the team communicated well. On a couple of occasions team members indicated reluctance to continue an ongoing discussion by focusing their attention on their screen and thereby withdrawing from the conversation. Furthermore, during an interview one team member shared an impression that discussions were sometimes very polite rather than being open and frank. Another team member said: ‘it is often easier to speak to people who agree with you most of the time. Otherwise you can spend a lot of time discussing.’ In a previous interview, this member said that different team members have different mindsets concerning the degree to which developers should be involved with requirements detailing. This indicates that some of the ratings for psychological distance may be due to cognitive distance and difficulties in reaching a common view on requirements.

7.1.4 Cognitive distance (M4)

Within the team, the cognitive distance between team members with different roles and length of experience was found to vary greatly. This is depicted in Fig. 4 where the maximum and average distance for each pair of team members is shown for the different aspects. The multiple measures used to assess these aspects of cognitive distance were combined to a normalised average value of 0.29 with the maximum value of 0.46, i.e. slightly below half the largest possible distance. Some of the measurements indicate long distances for their respective aspects. For example, there were large differences concerning technical skills in scope management (M4.2.1, 0.80 of 1) between the product owner and the tester. Long distances were also found between long-standing and new team members for knowledge of the local processes and organisation (M4.3), and the domain (M4.1, 0.80 of 1).

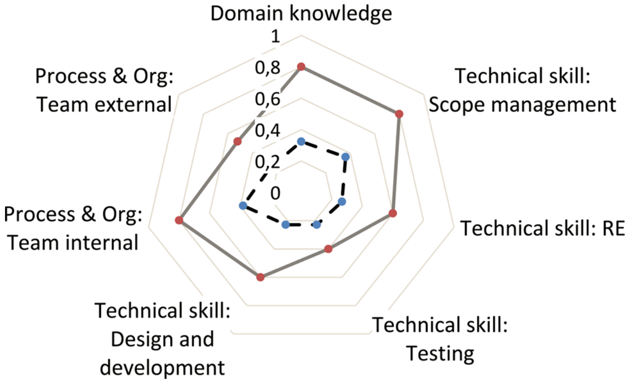

Fig. 4 Overview of the maximum cognitive distance found within team for the knowledge-based aspects, i.e. M4.1–M4.3. *Dashed line* shows the average distance within team

The team as a whole possesses near to the maximum amount of knowledge for the assessed aspects. Within the team there is *Expert* knowledge for the domain (M4.1), for local organisation and process (M4.3.1), and for 3 of the 4 technical skill areas (M4.4). Furthermore, for wider organisation and process (M4.3.2) and for testing (M4.2.4) there is *Proficient* knowledge. Furthermore, the team members on average have a high level of knowledge, around *Competent*, for all measured knowledge aspects (M4.1–4.3). One team member said: ‘I think we have a good mix of people who have been here a long time and new people.’ Another one said: ‘It is a good team! We’re well covered.’ This distribution can be a great asset for the team if the knowledge is utilised and shared in an efficient way.

Domain knowledge (M4.1) The tester said that he could be more proactive as his domain knowledge increased. A shorter cognitive distance between himself and the product owner and the requirements analyst enabled a faster response in the testing work, and quicker identification of issues. This correlates with the requirements analyst’s view that testers should think ‘outside the box’ and not just test according to agreed requirements. This requires testers to have good domain knowledge. Furthermore, a developer pointed out that the distance in domain knowledge between a very experienced requirements analyst and both newer developers and tester had on several occasions led to a failure to capture incorrect software behaviour. In these cases the requirements analyst had not communicated what he/she considered to be tacit requirements to the development team. These tacit requirements had then not been developed or tested, which was only discovered during user acceptance testing.

Technical skill (M4.2) The product owner and the requirements analyst both expressed that their previous experience of design and testing enables better requirements communication with the developers and testers. Furthermore, this enabled both the product owner and the requirements analyst to perform some user-related testing on the software.

Process and organisational knowledge (M4.3) One of the newer team members reported that he had little insight and knowledge of other teams and areas since there was limited interaction with them. However, this knowledge was increasing as time progressed through getting more involved in work at a wider project level and through more interactions with other teams. The scrum master indicated that synchronisation between teams could be improved by more frequent interactions and through sharing general information. The need for this was observed on several occasions when team members raised questions related to system testing, e.g. what requirements would be system tested, and how system testing issues were to be received by the development team.

Priorities (M4.4) The cognitive distance for priorities of individual system quality characteristics (M4.4) was low on average (normalised to <0.1) with no single characteristic scoring above 0.2, meaning that team members in general agreed on the relative priority for these characteristics, see Fig. 5. A larger distance was found when considering the total distance for priorities of all quality characteristics, on average 0.35 of 1 with a maximum distance of 0.6. In particular, these distances are longer between the product owner and other roles. For example, the requirements analyst prioritised functionality lower and maintainability higher than the product owner, while the tester prioritised usability much lower than all non-development roles including the product owner. This is surprising considering that the tester is responsible for verifying and validating the produced software, which in this case is an information system aimed at non-technical users.

There was agreement on the high importance of the quality characteristic *maintainability*; however, there were different reasons for this expressed by different roles. According to the product owner, requirements analyst and tester, this characteristic enabled the team to respond quickly to changing business requirements and bug reports. In contrast, the developers highlighted that maintainability was required because of the long life-expectancy of the system, and because most of the developers were on short-term contracts, thus the code is expected to be maintained by others in the future.

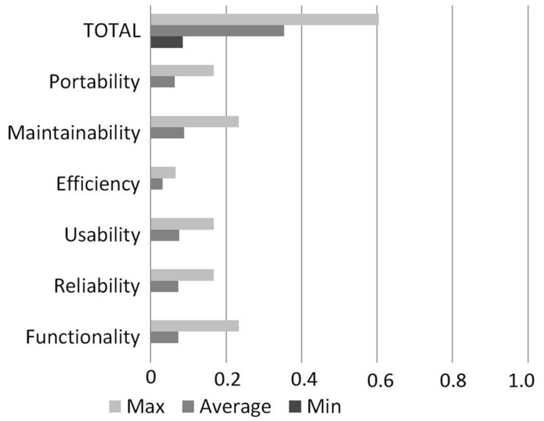

Fig. 5 Range of (normalised) cognitive distance for priorities between ISO quality characteristics between team members

There was a difference in viewpoint concerning the quality characteristics between developers and the other roles. One developer indicated that reliability, usability and efficiency were lower priority since these characteristics were mostly out of the control of this development team. Rather, these characteristics rely on software of lower-level architectural layers for which other teams are responsible.

Additional aspect of cognitive distance: agreed requirements (not currently covered by the iRE profile) The tester and the requirements analyst both suggested considering the difference, or distance, in knowledge of the agreed requirements. They had both observed that the testers who perform the user- and system-level testing may have very little knowledge (cognition) of the requirements implemented by the team. They described how testing without prior communication of the requirements resulted in large numbers of issues—but issues which later were rejected due to the software working according to the agreed requirements. Thus, there was a large distance in cognition of the agreed requirements between the system testers and the development team. At times, this distance was decreased through job rotation when a tester from the development team circulated to the system test team. In this case, the number of system test issues, which were rejected, was lower.

7.1.5 Adherence distance (M5)

Delivered versus agreed requirements (M5.1) The adherence distance between the agreed requirements and the behaviour of the delivered software was short. The product owner and the requirements analyst both stated that the delivered behaviour was *Almost the same* as the agreed requirements, while the tester judged that *Exactly the same* behaviour had been delivered as had been agreed. This indicates that the tester has a slightly different understanding of the agreed requirements compared to the product owner and the requirements analyst. Furthermore, the product owner said that at times the delivered behaviour was more and/or better than what had been agreed.

One developer said he had experienced that when there is a short distance in abstraction between the requirements and the software behaviour this detailed level of requirements restricts developer creativity. Similarly, the requirements analyst suggested that testing should go beyond the exact details of the agreed requirements in order to test and validate them effectively. Testing from this wider perspective might be encouraged by agreeing to requirements at a higher level of abstraction, thereby necessitating active consideration of the details by the developers and the testers.

Agreed versus documented requirements (M5.2) For the adherence distance between the documented and the agreed requirements, the interviewees saw no distance for the aspects of meaning (M5.2.1, Exactly the same) or coverage (M5.2.2, The same), and some distance for the aspect of abstraction level (M5.2.3). This distance in abstraction level is to be expected since documenting all details is infeasible. One interviewee pointed out that the requirements were documented at a significantly higher level of abstraction for agile than for traditional development. The normalised total average adherence distance between agreed requirements and requirements artefacts was 0.24 (M5.2), and this is solely due to the distance in abstraction levels.

Even though all respondents for this measurement (product owner, requirements analyst and tester) reported that the adherence distance was zero, two of them pointed out instances where the requirements artefact had not been updated after agreed scope changes. Thus there was a distance in meaning and coverage between the requirements artefacts and the current set of agreed requirements (M5.2.1 and M5.2.2).

7.1.6 Semantic distance (M6)

Some semantic distance was found between the requirements artefacts and the test artefacts. The meaning of the two artefacts (M6.1) was judged by the tester to be *Roughly the same*. This distance was mainly due to requirements information that was not yet in scope for the project, but already included in the documentation.

The tester judged that the test cases covered *Somewhat more* than what was specified in the requirements artefacts (M6.2). In addition, the abstraction level of the test cases (M6.3) was stated to be *Somewhat more* than for the requirements, which is to be expected. Furthermore, the tester commented that the level of detail in the test cases was dependent on time availability and was usually more than that provided in the specific set presented in the artefact questionnaire.

7.2 Practitioners' view on measuring distance

Throughout the study, feedback was gathered from the team members concerning their experience of measuring distances, both the general concept of measuring distance and the time and effort required of them to contribute to the measurements. These data were mainly gathered at the focus group, but also through the interviews and the observations.

At the focus group, the team members found the approach of distance measures useful in discussing issues and in identifying new areas for improvement. One workshop participant stated that the measurements unearthed new perspectives, e.g. concerning the psychological distance, and had enabled group reflection on previously un-discussed issues.

The team did not find the measurement of distances particularly costly. They had a high work load yet found time to complete the measurement questionnaires—approximately 10–15 min each. The scrum master also expressed that the team had not perceived any undue cost associated with participating in the study. The focus group was the most time-consuming part from their perspective, which took just over 60 min. The researchers' experience of the focus group was that presenting and reflecting on all the distance types in a satisfactory way required more time than was available.

The questions were understood by the participants and required no major clarifications once the measurement scales and the question on priority of quality characteristics (in the profile questionnaire) was explained. Minor clarifications were asked for, in particular for the questions on technical skills, because some participants found it hard to distinguish between technical knowledge and knowledge of company processes for a role.

7.3 Validity and limitations

We discuss the limitations of the results including threats to validity according to guidelines provided by Runeson et al. [65]. Steps taken to mitigate these limitations, and threats are also mentioned.

7.3.1 Construct validity

The main risk to construct validity is the precision of the distance measures. This risk concerns how well the measurement instruments assess the distances they are intended to measure. To mitigate this risk, the measurement instruments were designed in an iterative fashion based on empirical knowledge from previous studies, in combination with insight into the assessed case. Despite this, the construct validity of the measures requires further research to assess and improve on their precision. In fact, improvements to the measurements are part of the findings of this study. However, the main aim of this study was to perform a qualitative evaluation of the concept of distances, for which we judge that imprecisions of the measurements have a limited impact.

7.3.2 Internal validity

The main threat to internal validity is the risk of incorrectly assessing the impact of factors or missing impacting factors, and of participants misinterpreting the questions included in the distance measurement questionnaires. This is particularly relevant since we investigate relationships between RE distances and communication in a live context where there are multiple uncontrollable factors. This risk has been partly mitigated by studying one development team during a specific time period. Distances for a defined set of requirements and test cases, and for a specific group of people could then be investigated, thus enabling a study of how these people relate to each other and how they work with the specific requirements at hand. However, it remains an open risk that study participants and/or researchers have incorrectly identified factors, e.g. concerning the effect of an RE distance and that other relevant factors may have been missed.

Furthermore, there is a risk of participants having misunderstood questions, with subsequent impact on the obtained distance measurements. This risk was partly mitigated by obtaining knowledge of the case and designing the questions to match the applied processes and used terminology. In addition, the questions were reviewed and discussed with the second author who is more familiar with the case. Furthermore, triangulation was applied to the obtained distance measures by administering the questionnaires as part of an interview where misunderstandings could be discovered and resolved, and by comparing with data from the observations.

7.3.3 *External validity*

The question of external validity concerns the extent to which the results are applicable and of interest beyond that of the studied case, for which analytical generalisation needs to be considered. The set of distances investigated is based on a structured analysis of empirical data from another case (Company A, see 0). The fact that no conflicting findings have emerged when applying these to the case in this study indicates that the two cases are comparable when considering distances relevant to RET alignment. For this reason the results are of interest to cases displaying the characteristics common to the two cases on which these results are based (reported in Sect. 4 and Table 2): namely, small- and medium-sized co-located companies (150–200 people) and projects (10–20 people), with an agile and iterative development model, and for which there are no safety–critical aspects. In particular, more research is needed to determine the validity of the results for distributed development projects, i.e. projects with large geographical and temporal distances which often also entail socio-technical and power distances. Thus, generalizability needs to be considered on a case-by-case basis by comparing each case to the characteristics reported for this case.

There is a risk that the effects of distances found in this study are not applicable to projects using a different development model. In particular a strong focus on artefacts as the primary channel for requirements communication rather than face-to-face communication (as is the case for agile) might result in a different iRE profile from the one obtained in this study. However, results from a previous study show that even for a document-based process the degree of collaboration and thus distance between roles and individuals has a large impact on the collaboration between RE and later development activities and thus on the project outcome [8]. For this reason, the people-related distances are most likely relevant also for a traditional process model. Further research is required to explore the validity of these results for projects with a phase-based and document-based process.

7.3.4 *Reliability*

There is a risk that researcher biases have influenced the measurements and the interpretation of their impact and thus the reliability of the results. This risk was partly mitigated by including the perspectives of two researchers throughout the study and by applying triangulation to the collected data. For example, the research design and the measurement instruments were iteratively refined and reviewed by both researchers. Triangulation of the obtained distance measures was done by collecting further data on each distance through observations and interviews. Finally, the obtained distance measurements were validated by the development team at a focus group.

8 Findings and discussions

New insights into the impact of RE distances on requirements communication have been obtained in this study through measuring distances and investigating the impact of these for an ongoing development project. The three research questions can be answered based on the collected empirical data reported in the previous section. These focus on the effect of the distances on communication between RE and testing (RQ1), how the concept of distances can support the team's reflection (RQ2) and potential enhancements to the set of distances and the use of them (RQ3).

8.1 How RE distances affect requirements communication (RQ1)

All of the six investigated RE distances were found to affect the communication of requirements between RE and testing activities. No answer can be given for navigational

(D7) and temporal (D8) distance since they were excluded from the investigation (see Sect. 5.1.2). The development team had experienced the impact of people-related distances on requirements communication. However, for most distances the team members were previously not aware of the distance as such, but had merely observed its effects. This was the case for organisational distance, psychological distance and the priority aspect of cognitive distance. Highlighting these types of distances prompted the team members to consider them as potential causes of observed communication issues. Furthermore, the artefact-related distances were found to be indicators of requirements-test alignment rather than to have a direct impact on requirements communication. 0 summarises the impact of RE distances found for this case including the distances for which gaps were found, the team's prior awareness of each distance and the impact of these gaps. These findings are based on the observations of the team, on interviews and on experiences shared during the focus group session. We will now describe the details of the effects of the identified gaps for people-related and for artefact-related RE distances (Table 6).

8.1.1 The impact of people-related distances

According to the team members, all the people-related distances have an impact on requirements communication and agreement, both within the team and with roles outside the team such as system testers. Here we summarise the impacts and relate the findings to previous studies.

Misinterpreting and missing requirements is one consequence of long people-related distances, in particular *cognitive distances*. Within our case project, cognitive gaps concerning domain knowledge, led to a lack of communication of requirements on several occasions. This was true particularly for requirements which were tacit to requirements analysts with long experience, and not visible to developers and testers who had joined more recently. Similarly, a cognitive gap in the aspect of *priorities for the product* (M4.4) between individual team members and other roles contributed to missing quality requirements. Furthermore, a short cognitive distance can support communication and agreement of requirements details. In particular, a short cognitive distance (on testing knowledge) between both the requirements analyst and the product owner, and the tester can be beneficial since testing knowledge supports the requirements analyst in adapting the requirements information to the testers' needs.

Distances in cognition, or knowledge, between software engineers have not explicitly been studied before, as far as we are aware. However, there are some related findings within studies on domain knowledge and its impact on requirements. Based on previous research, Davis et al. [25] present a theory that simpler requirements elicitation techniques are sufficient when the user and the requirements engineer have similar domain knowledge, while elicitation is more challenging when their knowledge differs. Our findings confirm this, i.e. that cognitive gaps pose an RE communication challenge. Furthermore, Hofmann and Lehner [37] found that when there was a large difference in domain knowledge between the requirements engineer and the development team, this resulted in instances where unrealistic requirements were selected. In contrast, Fricker et al. [32] found that communication around the design between stakeholder and architects leads to shared understanding (cognition) of the requirements, and identification of tacit requirements and of needed requirements changes.

Delays and inefficiency in decision making are also negative effects of people-related distance. In particular, organisational distance (M2) between the product owner and the rest of the development team was described by several team members as leading to delays when there are disagreements concerning which requirements to implement. Whenever possible the team tries to resolve such issues internally rather than escalate them to their managers. Furthermore, team members expressed that difficulties experienced in communicating and agreeing within the team may be explained by psychological distance (M3.1 and M3.2).

These results confirm findings from our previous study of communications gaps where the organisational structure led to power struggles between different units and technical areas rather than to constructive communication on how to reach a common goal [8]. Similarly, Karlsson et al. [42] found a range of difficulties in coordinating marketing and development roles that include lack of common vocabulary and weak common views on the role and need of requirements details. Both of these difficulties occurring over organisational boundaries are related to cognitive distance, thus pointing to a possible correlation between organisational and cognitive distance. Since staff are often organised according to competence such a correlation could be expected, e.g. that development engineers are distant from business analysts in organisational terms, in knowledge and in cognition. Similarly, Curtis et al. reported that organizational boundaries can cause communication gaps that hinder the mutual understanding of requirements [20].

Delays in clarifying requirements and impeded coordination within the team were other effects of long *geographical distance* (M1) experienced in our case, in particular between the product owner and both the developers and tester. When the physical distance was shortened (soon after the focus group), the team experienced an increased frequency of communication with the product

Table 6 Summary of findings of how RE distances affect requirements communication (RQ1)

RE distance	Effect of gap	(G)ap identified. (A)wareness of gap within team. (I)mpact of gap observed
D1 Geographical	Delays and misunderstandings in requirements communication and coordination with the distant team member	GAI
D2 Organisational	Difficulties and delays in decision making concerning disagreements on which requirements to support	GI
D3 Psychological	Conflicts and difficulties in agreeing, e.g. when discussing requirements details	GI
D4 Cognitive	*Domain knowledge* missed communication of tacit requirements leading to identifying missing functionality at a late stage *Technical skill* (a) for *testing* and *development skills* a short distance supports good requirements communication towards developers and testers, and facilitates user-level testing (b) for scope management and requirements engineering: general impact on communication *Process and Organisational knowledge* (a) for *role of others* this distance can cause misalignment of system-level testing relative requirements delivered by team (b) for *own role* no direct impact was found *Priorities for system:* Missing quality requirements with subsequent misalignment of user expectations versus quality level in delivered software, may surface in system-level testing *Agreed requirements* Gaps concerning this (new) aspect between system testers and development team was suggested to lead to system-level testing of other non-agreed requirements with subsequent increase in potentially unnecessary issue reports and management of these	GA (domain) I (all)
D5 Adherence	Between delivered versus agreed requirements *Similarity and coverage* A long distance is a sign of misalignment between roles and activities in the development flow including missing or misunderstood communication of requirements and that the testing effort has failed to catch discrepancies between agreed and delivered requirements *Abstraction* A long distance concerning this (new) aspect may motivate developers and testers roles in validating requirements by providing more freedom and responsibility to detail them, which requires domain knowledge and insight into user expectations. Thus, a gap may in this case have a positive effect on the development process Between agreed versus documented requirements *Similarity and coverage:* a long distance can indicate either misalignment caused by missing or misunderstood requirements communication, or that the documentation is not updated *Abstraction*: the distance characterises the development model and in particular the weight given to requirements documentation, i.e. degree of documentation-based communication	I
D6 Semantic	*Similarity and coverage*: a gap can indicate that requirements are not fully updated or a misalignment in requirements communication towards testing *Abstraction*: a characteristics of the development model, i.e. degree of requirements detail and documentation produced upfront or concurrently with design and implementation	I

The following is shown for the studied case: G—the identified gaps, A—the team's awareness of the gap prior to the study, and I—the impact of the gap. Each bulleted item describes an aspect of distance

owner. They believe this decreased distance will contribute to reducing the amount of misunderstandings and misalignment with the users' needs and expectations.

That physical distance affects communication is well known and is researched within global software engineering [1] where it has been found to be a significant cost driver. Dibbern et al. [28] found that in cases where client-specific knowledge was crucial, face-to-face collaboration was required for adequate knowledge transfer of domain knowledge and for requirements analysis and specification.

Geographical distance also affects co-located development. This was reported by Allen in the 1970s. He found that a physical distance of just 25 metres reduces the communication between engineers [2]. Our results confirm that this is still relevant despite the use of new technology-based communication channels such as email, chat, application life-cycle managements systems etc.

The impact of geographical distance on requirements communication may be due to missing context and awareness [67]. Our finding that geographical distance impedes team coordination, e.g. that when the product owner is not readily available to clarify and discuss requirements the communication slows down, confirms previous results by Damian et al. [24] who found that awareness of one another's work affects the coordination within a project. This in turn leads to information sharing and knowledge gain but when awareness is missing this can lead to misunderstandings about requirements [22].

Further research of distances in the co-located context, especially for large development organisations, is needed to further understand how to design office space, development processes etc. in order to facilitate an improved flow of requirements.

8.1.2 Artefact-related distances as indicators

The artefact-related distances were found to be primarily indicators of project characteristics rather than factors that affect communication. Certain aspects of adherence distance were found to indicate: weak or strong alignment between RE and testing, which development model was being applied, and the up-to-datedness of an artefact. In general, while using metrics is acknowledged as a way of monitoring the status of a project or of process improvement [5] we are not aware of any research about using metrics for any of the characteristics in our study, i.e. RET alignment, the development model being used, or the degree to which artefacts are updated.

The degree of RET alignment for a project corresponds to the *adherence distance between delivered versus agreed requirements*. A discrepancy, or distance between these indicates that the testing effort has failed to catch unsupported requirements.

In our previous study, metrics were identified as an industrial practice for monitoring and gaining control of the alignment between RE and testing [10]. That study concluded that RET metrics enhance the awareness of the importance of alignment, thus increasing the incentives and motivation for applying good alignment practices [10]. Therefore, the concept of using distance measures as markers of RET alignment poses an interesting direction for future research.

The applied development model affects the *abstraction* aspect of *adherence distance* and *semantic distance* (M5.2.3, M6.3). The fact that our measure indicates a long distance for this is thus not judged to detect a gap, but rather as an indicator of the agile development model applied for the case. The distance in abstraction level between agreed and documented requirements is expected to be greater for an agile development project than for a project applying a traditional plan-driven development model. Subsequently, the adherence distance (M5.2) of a project will vary depending on how much weight is given to the requirements artefacts according to the applied development process.

Factors required for a successful agile deployment have been investigated by several researchers, e.g. Tsun [69], Misra [52], and Jalali [38], but we are not aware of any research that considers the underlying characteristics of development models. A theoretical model of such characteristics and how they affect the effectiveness of software practices could support organisations in configuring and adapting their software processes to their specific context and needs. For example, the abstraction level of the requirements relative to the test cases (adherence distance) could be adapted according to the differences in domain knowledge (cognitive distance) between the involved engineers.

Artefacts that have not been updated cause long *adherence distance* for the aspects of *similarity* and *coverage*, which thus indicates a discrepancy between documented and agreed requirements. In our case there was a gap in similarity and coverage both for adherence distance between agreed and documented requirements (M5.2.1 and M5.2.2) and for semantic distance between documented requirements and test cases (M6.1 and M6.2). However, this distance did not cause any gaps in the current requirements communication since the documented requirements merely supported the primary (face-to-face) communication channel. In contrast, for a project where the requirements artefact is the main source of requirements information requirements that have not been updated (indicated by adherence distance) are more likely to lead to miscommunication with the developers and testers, thus also affecting the implemented software.

Keeping the requirements specification updated is a known RE challenge [8, 42] that causes misunderstandings and rework within development projects. Charrada et al. developed and evaluated a method using natural language processing for identifying potential requirements that had not been updated based on code changes [17]. However, we are not aware of any research suggesting the use of measurements as indicators of how updated or outdated an artefact is. Further research is needed to investigate whether such measurements can be efficient and cost effective.

8.2 How RE distances support group reflection (RQ2)

The concept of distances was found to provide a good metaphor for discussing requirements communication within the development team. Presenting the distances and visualising the obtained measures at a focus group session stimulated group reflection around communication issues and enabled project members to identify areas for improvement.

The distance measures confirmed known issues, but also surfaced new issues. For example, several team members shared experiences of how the geographical distance to the product owner caused delays and misunderstandings. Similarly, the developers described how cognitive distance led to tacit requirements that were known to the product owners, but not to the developers and tester, to be missed and not discovered until customer acceptance testing. The distance concept provided an explanation for both of these communication gaps.

Furthermore, presenting measurements of psychological distance enabled the team to discuss issues that had been observed individually but never discussed within the team. Thus, the distance measurements enabled an objective discussion of what could otherwise be a sensitive topic. This confirms previous findings reported by Angermo Ringsted et al., namely that visual presentation of psychometrics can trigger group discussions of topics previously only noted by individuals [3].

Group reflection around distances can support teams in identifying practices that may improve alignment and communication. We observed that the conceptual image of bridging or shortening a problematic distance triggered the participants to suggest new improvement practices. This further illustrates the relevance of using distance as a metaphor when considering coordination within software development and concurs with the findings of Angermo Ringsted et al. [3] that objective measurements pointing to potential explanations can support teams in identifying improvements.

8.3 How the measurement and use of RE distances can be enhanced (RQ3)

We have identified a number of potential improvements related to two areas: the set of distances and how they are measured; and how to use them to enhance group reflection on coordination and communication issues.

8.3.1 The set of RE distances and how they are measured

Two cognitive distance aspects (D4) may be removed since no evidence of relevance was found through interviews, during the observations or the focus group session. These are technical skill (M4.2) and organisational & process knowledge (M4.3). In addition, two new aspects of distance were suggested during the interviews, namely knowledge of current requirements, and abstraction level of agreed requirements compared to delivered software. Knowledge of current requirements is an aspect of cognitive distance (D4) and concerns knowledge of what functionality and behaviour the software is intended to support. A long distance between the development team and the system test team may negatively affect RET alignment and result in unnecessary issue reports. In addition, a short distance for this aspect between all team members indicates a common view of the goal, a factor previously identified as supporting good communication [8] and RET alignment [10].

The other suggested new aspect relates to adherence distance (D5), namely *abstraction level of agreed requirements versus delivered software* that may affect test coverage and creativity. These were mentioned during interviews with the requirements analyst and one developer. For this aspect, a long distance could have positive effects. Namely that agreeing primarily on high-level requirements could encourage testers and developers to take on more responsibility and be more creative in detailing the requirements. This would enhance RET alignment by improving the validation and verification of requirements. However, this requires domain knowledge and insight into business strategies for those detailing the requirements.

As well as modifying the set of distances, the method for measuring distances can be improved, in particular for the artefact-related distances. We discovered that there is a high risk of bias when using self-assessment questionnaires for measuring adherence and semantic distances. This is because a long adherence or semantic distance indicates a failure to capture and document the agreed requirements—a fact that the responsible person is either not aware of or might be unwilling to acknowledge.

8.3.2 Group reflection on distances

Even though the presentation of distance measures supported group reflection, it was a challenge to select and visualise relevant distance measurements without overloading the participants. For our case, the focus group session ran out of time and the participants became tired, which we largely attribute to the amount of presented concepts and data. Group reflection on distance may be enhanced by further structuring the discussion to focus on sub-sets of distances and by improving the visualisation of distances. The meeting structure could be refined to present different sub-sets and categories of distance as relevant to the specific case. This may mitigate the problem of information overload and lack of time at the focus group meeting, and use meeting time more effectively. The categories of RE distances (level of RET alignment, degree of updatedness of artefacts and development model, see Sect. 8.1.2) could be used for this. Another set of distance measures to consider would be particularly key gaps, e.g. those along important communication paths.

Furthermore, improved visualisations of distance measurements and combinations of distances could support software engineers in analysing and comparing multiple aspects of distance. Visualisation techniques have been used in related research to support improvement discussions based on large amounts of data. For example, Feldt et al. [29] found that heat maps visualising multiple measurements can support the analysis and planning of quality assurance. These changes in how to present and discuss the distances are potential areas for future research.

9 Conclusions and future work

Communication is a basic pre-requisite for collaboration and coordination in general, and within software development requirements communication is a key tool for steering project members towards developing a product that meets the customers' expectations. Communication paths along which requirements flow can be mapped and analysed in order to identify and alleviate problems such as bottlenecks, missing or weak connections between individuals and roles. In our work we focus on the quality of the communication and consider how requirements communication is affected by distances. In our previous research we identified a set of RE distances between people and between artefacts. In this case study, measurement instruments for these distances have been designed and evaluated. The measurements were applied to an agile development project in order to investigate the impact of distance on communication with a focus on the coordination and alignment of RE and testing activities.

In this paper, we present new insights into RE distances concerning their relevance for requirements communication and coordination (RQ1), how the concept of distances supports group reflection of practices (RQ2), and how these measurements can be improved (RQ3). All of the RE distances were found to affect requirements communication and three main categories of distances were identified:

1. Those that can affect requirements communication,
2. Those that indicate weak or strong alignment, and
3. Those that characterise the applied development model, e.g. agile or plan-driven.

The metaphor of distance was found to support and stimulate group reflection on RET alignment and to enable development teams to identify new improvement areas. The concept of distances can also provide project teams with new perspectives and potential explanations of issues they experienced. In addition, providing objective measures of distance can support an open and objective discussion, even of more sensitive subjects such as individual difficulties in communicating.

The contributions of this study also include examples of measurement instruments for distances and the idea that a project's ability to coordinate and align can be gauged by assessing its level of RE integration, using an iRE profile. The iRE profile information can indicate the need for improved communication between distant roles, e.g. when there are long geographical, organisational and cognitive distances between an RE engineer and a tester this can be improved by practices such as regular meetings, requirements reviews, and good requirements documentation. The iRE profile can thus indicate potential weaknesses or gaps in the information flow. Furthermore, a project's RET alignment status may be monitored by re-producing its iRE profile on a regular basis, thus continuously assessing its integration level.

The measurement instrument and guidelines for applying it are available on-line [11] and may be used by practitioners to assess distances in software development projects. The case study findings can support development teams in reflecting on how RE distance may affect their requirements communication. The findings of this study can thus support teams with an increased awareness of the impact of distances on their communication and thereby facilitate identifying communication gaps and objectively discussing how to bridge distances causing these gaps.

Future work includes improving measurements for the artefact-related distances and further exploring the visualisation of distances and iRE profiles. An interesting avenue to explore is to identify patterns in iRE profiles between 'similar' projects, e.g. distributed projects, enterprise projects or large-scale projects. Future research into the use of distance measures as indicators of RET alignment has the potential of raising awareness of alignment within teams, both the current alignment status and the importance of alignment for successful software development.

In conclusion, RE distances affect the communication and decision making of requirements and the concept of distances can support teams in reflecting on their communication practices and improve their coordination in a novel way. Distance measurements allow practitioners to take a step back and consider underlying factors than cause problems rather than merely focusing on the issues themselves. Therefore, we conclude that RE distances have the potential to explain communication issues and to support improvements of communication practices.

Acknowledgments We would like to thank the development team members for enabling this study by sharing their time, thoughts and office space. We also want to thank the reviewers, both Dr Leonor Barroca and our anonymous reviewers for providing constructive feedback that has helped us improve the readability of our results. This work was partly funded by EASE http://ease.cs.lth.se and by Ericsson Research.

References

1. Agerfalk PJ, Fitzgerald B, Holmstrom Olsson H, Lings B, Lundell B, ÓConchúir E (2005) A framework for considering opportunities and threats in distributed software development. Proc Int Works Distr Softw Eng DiSD 2005:47–61
2. Allen T (1977) Managing the flow of technology. MIT Press, Cambridge, MA
3. Angermo Ringstad M, Dingsøyr T, Moe NB (2011) Agile process improvement: diagnosis and planning to improve teamwork. In: Proceedings of 18th European conference on systems, software and service process improvement (EuroSPI'11), Communications in computer and information science ,vol 172, pp 167–178
4. Barmi ZA, Ebrahimi AH, Feldt R (2011) Alignment of requirements specification and testing: a systematic mapping study. In: Proceedings of the 4th international conference on software testing, verification and validation workshops (ICSTW), pp 476–485
5. Basili VR, Rombach HD (1988) The TAME project: towards improvement-oriented software environments. IEEE Trans Softw Eng 14(6):758–773
6. Benner P (1982) From novice to expert. Am J Nurs 82(3): 402–407
7. Berntsson Svensson R, Aurum A, Paech B, Gorschek T, Sharma D (2012) Software architecture as a means of communication in a globally distributed software development context. In: Proceedings of the 13th international conference on product-focused software process improvement (PROFES'12), pp 175–189
8. Bjarnason E, Wnuk K, Regnell B (2011) Requirements are slipping through the gaps—a case study on cause and effects of communication gaps in large-scale software development. In: Proceedings of the 19th IEEE international requirements engineering conference, pp 37–46
9. Bjarnason E (2013) Distances between requirements engineering and later software development activities: a systematic map. In: Proceedings of requirements engineering for software quality conference (REFSQ) 2013, pp 292–307
10. Bjarnason E, Runeson P, Borg M et al (2013) Challenges and practices in aligning requirements with verification and validation: a case study of six companies. Empirical software engineering, published on-line July 2013
11. Bjarnason E (2013) Research material for Gap Finder evaluation study incl measurement instruments, interview guide etc. (latest access: 2015-02-27). http://serg.cs.lth.se/research/experiment_packages/GapFinder/
12. Bjarnason E, Smolander K, Engström E, Runeson P (2014) Alignment practices affect distances in software development: a theory and a model. 3rd Semat workshop on general theories of software engineering (GTSE14), pp 21–31
13. Bjarnason E, Hess A, Berntsson Svensson R, Regnell B, Doerr J (2014) Reflecting on evidence-based timelines. IEEE Softw 31(4):37–43
14. Briand LC, Labiche Y, O'Sullivan L, Sówka MM (2006) Automated impact analysis of UML models. J Syst Softw 79(3):339–352
15. Calefato F, Damian D, Lanubile F (2007) An empirical investigation on text-based communication in distributed requirements workshops. In: Proceedings of 2nd international conference on global software engineering (ICGSE 2007), pp 3–11. doi:10.1109/ICGSE.2007.9
16. Cataldo M, Herbsleb J, Carley K (2008) Socio-technical congruence: a framework for assessing the impact of technical and work dependencies on software development productivity. In: Proceedings of 2nd ACM-IEEE international symposium on empirical software engineering and measurements (ESEM '08)
17. Charrada EB, Koziolek A, Glinz M (2012) Identifying outdated requirements based on source code changes. In: 20th IEEE international requirements engineering conference (RE), pp 61–70
18. Cheng BH, Atlee JM (2007) Research directions in requirement engineering. In: Proceeding of future of software engineering (FOSE), pp 285–303, May 2007
19. Cleland-Huang J, Chang CK, Christensen M (2003) Event-based traceability for managing evolutionary change. IEEE Trans Softw 29(9):796–810
20. Curtis B, Krasner H, Iscoe N (1988) A field study of the software design process for large systems. Commun ACM 31(11): 1268–1287
21. Damian D (2001) An empirical study of requirements engineering in distributed software projects: is distance negotiation more effective? In: Proceedings of 8th Asia-Pacific software engineering conference (APSEC 2001), pp 149–152
22. Damian DE, Zowghi D (2003) Requirements engineering challenges in multi-site software development organizations. Req Eng J 8:149–160
23. Damian D, Chisan J, Vaidyanathasamy L, Pal Y (2005) Requirements engineering and downstream software development: findings from a case study. Empir Softw Eng 10:255–283
24. Damian D, Kwan I, Marczak S (2010) Requirements-driven collaboration: leveraging the invisible relationships between requirements and people. Collaborative software engineering. Springer, New York, pp 57–76
25. Davis CJ, Tremblay MC, Fuller RM, Berndt DJ (2006) Communication challenges in requirements elicitation and use of the repertory grid technique. J Comput Inf Syst 46(5):78 (Special Issue 2006)
26. De Lucia A, Fasano F, Oliveto R, Tortora G (2007) Recovering traceability links in software artifact management systems using information retrieval methods. ACM Trans Softw Eng Methodol 16(4):Article 13
27. Dias Neto AC, Arilo C, Subramanyan R, Vieira M, Travassos GH (2007) A survey on model-based testing approaches: a systematic review. In: Proceedings of 1st ACM international workshop on empirical assessment of software engineering languages and technologies, pp 31–36
28. Dibbern J, Winkler J, Heinz A (2008) Explaining variations in client extra costs between software projects offshored to India. MIS Q Manag Inf Syst 32(2):333–366
29. Feldt R, Staron M, Hult E, Liljegren T (2013) Supporting software decision meetings: heatmaps for visualising test and code measurements. In: Software engineering and advanced applications (SEAA), 39th EUROMICRO conference on, pp 62–69
30. Feldt R, Torkar R, Lefteris A, Samuelsson M (2008) Proceedings of 2008 international workshop on cooperative and human aspects of software engineering (CHASE'08), pp 49–52

31. Flemming WR (1978) Requirements communication. International conference automatic testing (AUTOMTESTCON'78), pp 228–229. doi: 10.1109/AUTEST.1978.764370
32. Fricker S, Glinz M (2010) Comparison of requirements hand-off, analysis, and negotiation: case study. In: Proceedings of IEEE international requirements engineering conference, Sept 2010, pp 167–176. doi:10.1109/RE.2010.29
33. Gotel O, Finkelstein A (1994) An analysis of the requirements traceability problem. In: Proceedings of first international conference on requirements engineering, pp 94–101
34. Grieskamp W, Kicillof N, Stobie K, Braberman V (2011) Model-based quality assurance of protocol documentation: tools and methodology. Softw Test Verif Reliab 21(1):55–71
35. Hasling B, Goetz H, Beetz K (2008) Model based testing of system requirements using UML use case models. In: Proceedings of 2008 international conference on software testing, verification, and validation
36. Hayes JH, Dekhtyar A, Sundaram SK, Holbrook EA, Vadlamudi S, April A (2007) REquirements TRacing on target (RETRO): improving software maintenance through traceability recovery. Innov Syst Softw Eng 3(3):193–202
37. Hofmann HF, Lehner F (2001) Requirements engineering as a success factor in software projects. IEEE Softw 18(4):58–66
38. Jalali S, Gencel C, Šmite D (2010) Trust dynamics in global software engineering. In: Proceedings of the 2010 ACM-IEEE international symposium on empirical software engineering and measurement. ACM, New York, p 23
39. Jarke M (1998) Requirements traceability. Comm ACM 41(12):32–36
40. Jilani LL, Desharnais J, Mili A (2001) Defining and applying measures of distance between specifications. IEEE Trans Softw Eng 27(8):673–703
41. Jørgensen M, Sjøberg D (2000) The importance of NOT learning from experience. Proceedings of European software process improvement, EuroSPI'2000, pp 2.2–2.8
42. Karlsson L, Dahlstedt AG, Regnell B, Natt och Dag J, Persson A (2007) Requirements engineering challenges in market-driven software development-an interview study with practitioners. Inf Softw Technol 49(6):588–604
43. Kraut RE, Streeter L (1995) Coordination in software development. Commun ACM 38(3):69–81
44. Kukkanen J, Vakevainen K, Kauppinen M, Uusitalo E (2009) Applying a systematic approach to link requirements and testing: a case study. In: Proceedings of Asia-Pacific software engineering conference (APSEC '09):482–488
45. Kwan I, Marczak S, Damian D (2007) Viewing project collaborators who work on interrelated requirements. In: Proceedings of 15th IEEE international requirements engineering conference (RE'07), pp 369–370
46. Lormans M, van Deursen A, Gross H (2008) An Industrial case study in reconstructing requirements views. Emp Softw Eng 13(6):727–760
47. Lubars M, Potts C, Richter C (1993) A review of the state of the practice in requirements modelling. In: Proceedings of 1st IEEE international symposium on requirements engineering, pp 2–14
48. Marczak S, Damian D, Stege U, Schröter A (2008) Information brokers in requirement-dependency social networks. In: Proceedings of 16th IEEE international conference on requirements engineering (RE'08), pp 53–62
49. Marczak S, Damian D (2011) How interaction between roles shapes the communication structure in requirements-driven collaboration. In: 19th IEEE international requirements engineering conference, pp 47–56
50. Martin R, Melnik G (2008) Tests and requirements, requirements and tests a Möbius strip. IEEE Softw 25(1):54–59
51. Melnik G, Maurer F, Chiasson M (2006) Executable acceptance tests for communicating business requirements: customer perspective. In: Proceedings of agile conference, Minneapolis, USA, pp 12–46
52. Misra SH, Kumar V, Kumar U (2009) Identifying some important success factors in adopting agile software development practices. J Syst Softw 82(11):1869–1890
53. Mohagheghi P, Dehlen V (2008) Where is the proof? A review of experiences from applying MDE in industry. In: Proceedings of model driven architecture–foundations and applications, pp 432–443
54. Moe NB, Dingsøyr T, Røyrvik EA (2009) Putting agile teamwork to the test—an preliminary instrument for empirically assessing and improving agile software development. In: Proceedings of XP 2009, LNBIP 31, pp 114–123
55. Nebut C, Fleurey F, Traon YL, Jézéquel J (2006) Automatic test generation: a use case driven approach. IEEE Trans Softw Eng 32(3):140–155
56. Post H, Sinz C, Merz F, Gorges T, Kropf T (2009) Linking functional requirements and software verification. In: Proceedings of 17th IEEE international requirements engineering conference, pp 295–302
57. Ramesh B, Stubbs C, Powers T, Edwards M (1997) Requirements traceability: theory and practice. Ann Softw Eng 3(1):397–415
58. Ramesh B (1998) Factors influencing requirements traceability practice. Commun ACM CACM Homepage Arch 41(12):37–44
59. Ramesh B, Cao L, Baskerville R (2010) Agile requirements engineering practices and challenges: an empirical study. Inf Syst J 20(5):449–480
60. Randell B (1969) Towards a methodology of computing system design. NATO working conference on software engineering 1968, report on a conference sponsored by NATO Scientific Committee, Garmisch, Germany, pp 204–208
61. Regnell B, Runeson P (1998) Combining Scenario-based requirements with static verification and dynamic testing. In: Proceedings of 4th international working conference requirements engineering: foundation for software quality, pp 195–206
62. Regnell B, Runeson P, Wohlin C (2000) Towards integration of use case modelling and usage-based testing. J Syst Softw 50(2):117–130
63. Robinson H, Segal J, Sharp H (2007) Ethnographically-informed empirical studies of software practice. Inf Softw Technol 49:540–551
64. Robson C (2002) Real world research, 2nd ed. Blackwell Publishing, Hoboken
65. Runeson P, Höst M, Rainer A, Regnell B (2012) Case study research in software engineering—guidelines and examples. Wiley, Hoboken
66. Sabaliauskaite G, Loconsole A, Engström E, Unterkalmsteiner M, Regnell B, Runeson P, Gorschek T, Feldt R (2010) Challenges in aligning requirements engineering and verification in a large-scale industrial context. In: Proceedings of REFSQ 2010
67. Stapel K, Knauss E, Schneider K (2009) Using FLOW to improve communication of requirements in globally distributed software projects. In: IEEE proceedings of international workshop on collaboration and intercultural issues on requirements: communication, understanding and softskills, pp 5–14
68. Stapel K, Knauss E, Schneider K, Zazworka N (2011) FLOW mapping: planning and managing communication in distributed teams. In: Proceedings of 6th IEEE international conference on global software engineering (ICGSE), pp 190–199
69. Tsun C, Dac-Buu C (2008) A survey study of critical success factors in agile software projects. J Syst Softw 81(6):961–971
70. Uusitalo EJ, Komssi M, Kauppinen M et al (2008) Linking requirements and testing in practice. In: 16th IEEE international requirements engineering conference, NJ, USA, pp 265–270

71. Watkins R, Neal M (1994) Why and how of requirements tracing. IEEE Softw 11(4):104–106
72. Wolf T, Nguyen T, Damian D (2008) Does distance still matter? J Impr Practice Softw Process 13(6):493–510
73. Yue T, Briand LC, Labiche Y (2011) A systematic review of transformation approaches between user requirements and analysis models. Requir Eng 16(2):75–99
74. von Zedtwitz M (2002) Organizational learning through post-project reviews in R&D. R&D Manag 32(3):255–268